

REMARKS

This is a full and timely response to the outstanding non-final Office Action mailed August 13, 2004. Reconsideration and allowance of the application and pending claims are respectfully requested.

I. Abstract Objection

The abstract of the disclosure has been objected to for allegedly not including the aspects of the invention that are new in the art. Although Applicant disagrees with this viewpoint for reasons discussed below, Applicant has amended the abstract to provide further detail as to an embodiment of Applicant's invention to expedite issuance of a patent. Applicant respectfully requests that the objection be withdrawn.

II. Specification Objection

The specification has been objected to because, it is alleged, the title of the invention is not descriptive. Applicant disagrees with this position. Specifically, the title indicates that the patent application describes systems and methods for "dynamically replacing code." As is described in detail in the specification, such dynamic code replacement refers to replacing code while a program that contains the code to be replaced is executing (as opposed to static code replacement; see, e.g., Donohue). Applicant therefore believes that the title is descriptive of the claimed inventions and therefore has not amended the title of the patent application.

III. Claim Rejections - 35 U.S.C. § 112, Second Paragraph

Claim 7 has been rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter

which the Applicant regards as the invention. In particular, the Office Action states that the term “dynamic execution layer interface” is not an art-recognized term and the meaning of the term is not particularly pointed out or distinctly defined in either the specification or the claims.

While Applicant disagrees (the specification describes the dynamic execution layer interface (DELI) and what that interface does in great detail), Applicant has amended claim 7 to generically recite a “software interface.” Applicant submits that this term is well known in the relevant art. In view of that amendment, it is respectfully asserted that claim 7 defines the invention in the manner required by 35 U.S.C. § 112. Accordingly, Applicant respectfully requests that the rejection to claim 7 be withdrawn.

IV. Claim Rejections - 35 U.S.C. § 102(a)

Claims 1-22 have been rejected under 35 U.S.C. § 102(a) as being anticipated by Donohue (U.S. Pat. No. 6,202,207). Applicant respectfully traverses this rejection.

It is axiomatic that “[a]nticipation requires the disclosure in a single prior art reference of each element of the claim under consideration.” W. L. Gore & Associates, Inc. v. Garlock, Inc., 721 F.2d 1540, 1554, 220 U.S.P.Q. 303, 313 (Fed. Cir. 1983)(emphasis added). Therefore, every claimed feature of the claimed invention must be represented in the applied reference to constitute a proper rejection under 35 U.S.C. § 102(a).

In the present case, not every feature of the claimed invention is represented in the Donohue reference. Applicant discusses the Donohue disclosure and the applicability of that disclosure to Applicant’s claims in the following.

Donohue discloses a method and mechanism for synchronized updating of interoperating software. As is described by Donohue (column 5, lines 54-62):

An updater component according to a preferred embodiment of the invention thus controls upgrading of, and fixing of bugs within, an associated software product or products automatically without requiring any interaction by the user after an initial agreement of update criteria. The update criteria can be associated with the products' licensing terms and conditions. This ensures that users who adopt a suitable update policy can always have the most up-to-date software available, with errors being corrected automatically from the viewpoint of the user. The user does not need to know where software updates come from, how to obtain them or how to install them since the update component takes care of this. The software vendor avoids having to ship special CDs or diskettes to correct errors or provide additional features; the vendor can easily release code on an incremental basis such that customers receive new product features sooner and with no effort.

As is further described by Donohue (column 6, lines 11-35):

A system according to the invention preferably retrieves update resources for implementing an update of an installed computer program by sending an update request to one or more computer systems at network locations at which the required software resources are provided (together with a list of update resources and pre-requisites). The information used to identify the network locations may be explicit network location information held by the updater component, or it may be a software vendor name or any other information which can be used as a search parameter for identifying the location. In the preferred embodiment, the information is a product identifier which is provided by the updater component to a network search engine to initiate a search to identify the relevant network location at which are stored software resources for implementing an update to that product. This

search may be performed by a conventional Internet (or other network) search engine which is called by the updater component. When the search engine returns an identification of the network location, the updater component retrieves from this location a list of available relevant updates, checks the list against the locally held software product version and against predefined update criteria, and retrieves the update resources onto the local computer system if those criteria are satisfied.

As is readily apparent from the above-provided excerpts, the Donohue system is a *static* code replacement system such as that described in Applicant's Background of the Invention. As is provided in that section of Applicant's specification, such static code replacement can be undesirable. Specifically, Applicant states (page 1, line 14 to page 2, line 3):

Although it is not necessarily difficult to patch software in the manner described above, such patching is static. Specifically, the software patch must be developed off-line and installed while the application is not running. This can create problems where the application is one that must run continuously, *e.g.*, network server applications, financial transaction applications, telephone switching applications, airline reservation and air traffic control system applications, *etc.* When it comes to such applications, the user must be able to upgrade the software to fix bugs, improve performance, expand functionality, and so forth. In the simplest case, upgrades and bug fixes require the system to be shut down, updated, and then brought back on-line. This, of course, is not acceptable for non-stop applications and, at best, will result in loss of service and revenue.

In contrast to such static code replacement, Applicant claims *dynamic* code replacement in which code is replaced during execution of the program containing the

code to be replaced. For example, as provided in independent claim 1, Applicant claims (emphasis added):

1. A method for *dynamically patching code*, comprising the steps of:
 - intercepting original program instructions *during execution of the program*;
 - determining if an original program instruction is to be replaced;
 - and
 - dynamically replacing the original program instruction* with a replacement instruction by *fetching the replacement instruction and storing the replacement instruction in a code cache from which the replacement instruction can be executed in lieu of the original program instruction*.

Donohue clearly does not teach or suggest such “dynamic” patching of code. Moreover, Donohue does not teach or suggest “intercepting original program instructions during execution of the program”. As a first matter, Donohue does not even describe *any* “intercepting” of program instructions. Second, even if such intercepting were taught, no intercepting of program instructions “during execution of the program” is anticipated by Donohue.

In view of the above, it is clear that Donohue does not anticipate claim 1, or the claims that depend therefrom. Given that each of the other independent claims also contains limitations regarding interception of original program instructions during program execution, Applicant notes that the other claims that remain in the application are likewise not anticipated by Donohue.

Applicant further notes that Donohue fails to teach or suggest “dynamically replacing the original program instruction with a replacement instruction by fetching

the replacement instruction and storing the replacement instruction in a code cache from which the replacement instruction can be executed in lieu of the original program instruction”, as is also required by claim 1. First, as is noted above, Donohue does not “dynamically” replace code. All code replacement contemplated by Donohue is static. Second, Donohue does not contemplate fetching a replacement instruction and “storing the replacement instruction in a code cache from which the replacement instruction can be executed in lieu of the original program instruction”. Donohue therefore fails to anticipate claim 1 and its dependents for these reasons. Furthermore, given that each of the other independent claims also contain limitations regarding dynamically replacing the original program instructions and fetching replacement code and storing it in a code cache from which the replacement code can be executed in lieu of the original program instructions, Applicant notes that the other claims that remain in the application are likewise not anticipated by Donohue.

Due to the shortcomings of the Donohue reference described in the foregoing, Applicant respectfully asserts that Donohue does not anticipate Applicant’s claims. Therefore, Applicant respectfully requests that the rejection of these claims be withdrawn.

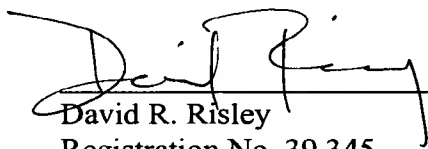
V. Canceled Claims

As identified above, claims 2, 11, 15, and 19 have been canceled from the application through this Response without prejudice, waiver, or disclaimer. Applicant reserves the right to present these canceled claims, or variants thereof, in continuing applications to be filed subsequently.

CONCLUSION

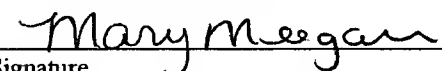
Applicant respectfully submits that Applicant's pending claims are in condition for allowance. Favorable reconsideration and allowance of the present application and all pending claims are hereby courteously requested. If, in the opinion of the Examiner, a telephonic conference would expedite the examination of this matter, the Examiner is invited to call the undersigned attorney at (770) 933-9500.

Respectfully submitted,


David R. Risley
Registration No. 39,345

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail, postage prepaid, in an envelope addressed to: Assistant Commissioner for Patents, Alexandria, Virginia 22313-1450, on

9-15-04


Signature